

IN THE CLAIMS:

Please cancel without prejudice the non-elected claims 19-21, 40-42 and 61-63, amend claims 1, 22, 43, 64, 86, 91 and 93-98, and add new claims 99-107, as indicated in the complete listing of claims provided below.

1. (currently amended) A method of managing file extensions in a digital processing system with a user interface and a plurality of files, each file having a name that comprises a filename and an extension, said method comprising:
associating a file with an indicator which is user selectable for a single file in the plurality of files in said digital processing system and which indicates how to display an extension of the file; and
displaying a displayed name of the file in the user interface in a style determined by said indicator.
2. (original) A method as in claim 1 wherein the style is such that the displayed name contains the extension of the file only when said indicator is for showing the extension of the file.
3. (original) A method as in claim 2 wherein said indicator is a bit, a file, an entry in a file, or an entry in a database, wherein said indicator in one state indicates hiding the extension and said indicator in another state indicates showing the extension.
4. (original) A method as in claim 3 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.

5. (original) A method as in claim 3 further comprising:
updating said indicator in response to an input event.
6. (original) A method as in claim 5 wherein the input event is that a new name is
specified in the user interface for the file.
7. (original) A method as in claim 6 wherein if the new name contains no extension,
then said indicator is set to hide the extension of the file in the user interface.
8. (original) A method as in claim 7 wherein only the filename of the file is replaced by
the new name so that the extension of the file is not changed.
9. (original) A method as in claim 6 wherein if the new name contains no extension and
the extension of the file is an empty string, then said indicator is set to a state that
takes a minimum amount of memory to store said state.
10. (original) A method as in claim 6 wherein if the new name comprising an extension
and a filename, then said indicator is set to show the extension of the file in the user
interface.
11. (original) A method as in claim 10 wherein the filename of the file and the extension
of the file are replaced by the filename of the new name and the extension of the new
name.


12. (original) A method as in claim 3 further comprising:
detecting if a first file that has a first filename and a first extension has a naming
conflict with a second file that has a second filename and a second extension,
wherein said first file has a first displayed name in the user interface and said
second file has a second displayed name in the user interface.
13. (original) A method as in claim 12 wherein if the first displayed name is the same as
the second displayed name, then a naming conflict is detected.
14. (original) A method as in claim 12 wherein if the first filename and the first extension
are the same as the second filename and the second extension, then a naming conflict
is detected.
15. (original) A method as in claim 1 further comprising:
exporting both the filename of the file and the extension of the file to a remote system
when the file is transferred to the remote system.
16. (original) A method as in claim 15 further comprising:
exporting said indicator to the remote system when the file is transferred to the remote
system.
17. (original) A method as in claim 1 further comprising:
importing both the filename of the file and the extension of the file from a remote
system when the file is transferred from the remote system.

18. (original) A method as in claim 17 further comprising:
importing said indicator from the remote system when the file is transferred from the
remote system.
- 19-21. (canceled)
22. (currently amended) A machine readable media for use with a digital processing
system which has a user interface and a plurality of files, each file having a name and
an extension, said machine readable media containing executable computer program
instructions which when executed by said digital processing system causes said
system to perform a method comprising:
associating a file with an indicator which is user selectable for a single file in the
plurality of files in said digital processing system and which indicates how to
display an extension of the file; and
displaying a displayed name of the file in the user interface in a style determined by
said indicator.
23. (original) A media as in claim 22 wherein the style is such that the displayed name
contains the extension of the file only when said indicator is for showing the
extension of the file.
24. (original) A media as in claim 23 wherein said indicator is a bit, a file, an entry in a
file, or an entry in a database, wherein said indicator in one state indicates hiding the
extension and said indicator in another state indicates showing the extension.

25. (original) A media as in claim 24 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.
26. (original) A media as in claim 24 wherein the method further comprises:
updating said indicator in response to an input event.
27. (original) A media as in claim 26 wherein the input event is that a new name is specified in the user interface for the file.
28. (original) A media as in claim 27 wherein if the new name contains no extension, then said indicator is set to hide the extension of the file in the user interface.
29. (original) A media as in claim 28 wherein only the filename of the file is replaced by the new name so that the extension of the file is not changed.
30. (original) A media as in claim 27 wherein if the new name contains no extension and the extension of the file is an empty string, then said indicator is set to a state that takes a minimum amount of memory to store said state.
31. (original) A media as in claim 27 wherein if the new name comprising an extension and a filename, then said indicator is set to show the extension of the file in the user interface.

32. (original) A media as in claim 31 wherein the filename of the file and the extension of the file are replaced by the filename of the new name and the extension of the new name.
33. (original) A media as in claim 24 wherein the method further comprises:
detecting if a first file that has a first filename and a first extension has a naming conflict with a second file that has a second filename and a second extension, wherein said first file has a first displayed name in the user interface and said second file has a second displayed name in the user interface.
34. (original) A media as in claim 33 wherein if the first displayed name is the same as the second displayed name, then a naming conflict is detected.
35. (original) A media as in claim 33 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a naming conflict is detected.
36. (original) A media as in claim 22 wherein the method further comprises:
exporting both the filename of the file and the extension of the file to a remote system when the file is transferred to the remote system.
37. (original) A media as in claim 36 wherein the method further comprises:
exporting said indicator to the remote system when the file is transferred to the remote system.

38. (original) A media as in claim 22 wherein the method further comprises:
importing both the filename of the file and the extension of the file from a remote
system when the file is transferred from the remote system.
39. (original) A media as in claim 38 wherein the method further comprises:
importing said indicator from the remote system when the file is transferred from the
remote system.
- 40-42. (canceled)
43. (currently amended) A digital processing system with a user interface and a plurality
of files, each file having a name that comprises a filename and an extension, said
system comprising:
means for associating a file with an indicator which is user selectable for a single file
in the plurality of files in said digital processing system and which indicates
how to display extensions of the files; and
means for displaying a displayed name of the file in the user interface in a style
determined by said indicator.
44. (original) A system as in claim 43 wherein the style is such that the displayed name
contains the extension of the file only when said indicator is for showing the
extension of the file.

- 
45. (original) A system as in claim 44 wherein said indicator is a bit, a file, an entry in a file, or an entry in a database, wherein said indicator in one state indicates hiding the extension and said indicator in another state indicates showing the extension.
46. (original) A system as in claim 45 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.
47. (original) A system as in claim 45 further comprising:
means for updating said indicator in response to an input event.
48. (original) A system as in claim 47 wherein the input event is that a new name is specified in the user interface for the file.
49. (original) A system as in claim 48 wherein if the new name contains no extension, then said indicator is set to hide the extension of the file in the user interface.
50. (original) A system as in claim 49 wherein only the filename of the file is replaced by the new name so that the extension of the file is not changed.
51. (original) A system as in claim 48 wherein if the new name contains no extension and the extension of the file is an empty string, then said indicator is set to a state that takes a minimum amount of memory to store said state.

52. (original) A system as in claim 48 wherein if the new name comprising an extension and a filename, then said indicator is set to show the extension of the file in the user interface.
53. (original) A system as in claim 52 wherein the filename of the file and the extension of the file are replaced by the filename of the new name and the extension of the new name.
54. (original) A system as in claim 45 further comprising:
means for detecting if a first file that has a first filename and a first extension has a naming conflict with a second file that has a second filename and a second extension, wherein said first file has a first displayed name in the user interface and said second file has a second displayed name in the user interface.
55. (original) A system as in claim 54 wherein if the first displayed name is the same as the second displayed name, then a naming conflict is detected.
56. (original) A system as in claim 54 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a naming conflict is detected.
57. (original) A system as in claim 43 further comprising:

means for exporting both the filename of the file and the extension of the file to a remote system when the file is transferred to the remote system.

58. (original) A system as in claim 57 further comprising:
means for exporting said indicator to the remote system when the file is transferred to the remote system.

59. (original) A system as in claim 43 further comprising:
means for importing both the filename of the file and the extension of the file from a remote system when the file is transferred from the remote system.

60. (original) A system as in claim 59 further comprising:
means for importing said indicator from the remote system when the file is transferred from the remote system.

61-63. (canceled)

64. (currently amended) A processing system comprising:
a processor;
a display device coupled to said processor, said display device displaying a user interface; and
a memory coupled to said processor, said memory storing a plurality of files, each file having a name that comprises a filename and an extension, said memory storing an indicator for a file which is user selectable for a single file in said plurality of files and which indicates how to display an extension associated

with the file, said processor displaying a displayed name of said file in said user interface in a style determined by said indicator.

65. (original) A processing system as in claim 64 wherein the style is such that the displayed name contains the extension of the file only when said indicator is for showing the extension of the file.
66. (original) A processing system as in claim 65 wherein said indicator is a bit, a file, an entry in a file, or an entry in a database, wherein said indicator in one state indicates hiding the extension and said indicator in another state indicates showing the extension.
67. (original) A processing system as in claim 66 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.
68. (original) A processing system as in claim 66 further comprising:
an input device coupled with said processor, said processor updating said indicator in response to an input event detected by said input device.
69. (original) A processing system as in claim 68 wherein the input event is that a new name is specified in the user interface for the file.

70. (original) A processing system as in claim 69 wherein if the new name contains no extension, then said indicator is set to hide the extension of the file in the user interface.
71. (original) A processing system as in claim 70 wherein only the filename of the file is replaced by the new name so that the extension of the file is not changed.
72. (original) A processing system as in claim 69 wherein if the new name contains no extension and the extension of the file is an empty string, then said indicator is set to a state that takes a minimum amount of memory to store said state.
73. (original) A processing system as in claim 69 wherein if the new name comprising an extension and a filename, then said indicator is set to show the extension of the file in the user interface.
74. (original) A processing system as in claim 73 wherein the filename of the file and the extension of the file are replaced by the filename of the new name and the extension of the new name.
75. (original) A processing system as in claim 66 wherein said processor detects if a first file that has a first filename and a first extension has a naming conflict with a second file that has a second filename and a second extension, wherein said first file has a first displayed name in the user interface and said second file has a second displayed name in the user interface.

76. (original) A processing system as in claim 75 wherein if the first displayed name is the same as the second displayed name, then a naming conflict is detected.
77. (original) A processing system as in claim 75 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a naming conflict is detected.
78. (original) A processing system as in claim 64 further comprising:
a network interface coupled to the said processor, said processor exports both the
filename of the file and the extension of the file to a remote system when the
file is transferred to the remote system through said network interface.
79. (original) A processing system as in claim 78 wherein said processor exports said
indicator to the remote system when the file is transferred to the remote system
through said network interface.
80. (original) A processing system as in claim 64 further comprising:
a removable memory coupled to the said processor, said processor exports both the
filename of the file and the extension of the file to a remote system when the
file is transferred to the remote system through said removable memory.
81. (original) A processing system as in claim 80 wherein said processor exports said
indicator to the remote system when the file is transferred to the remote system
through said removable memory.

82. (original) A processing system as in claim 64 further comprising:
a network interface coupled to the said processor, said processor imports both the
filename of the file and the extension of the file from a remote system when
the file is transferred from the remote system through said network interface.
83. (original) A processing system as in claim 82 wherein said processor imports said
indicator from the remote system when the file is transferred from the remote system
through said network interface.
84. (original) A processing system as in claim 64 further comprising:
a removable memory coupled to the said processor, said processor imports both the
filename of the file and the extension of the file from a remote system when
the file is transferred from the remote system through said removable memory.
85. (original) A processing system as in claim 84 wherein said processor imports said
indicator from the remote system when the file is transferred from the remote system
through said removable memory.
86. (currently amended) A processing system comprising:
a processor;
a display device coupled to said processor, said display device displaying a user
interface; and
a memory coupled to said processor, said memory storing in a file container a first file
which has a first extension and a first filename, said memory storing in said

file container a second file which has a second filename and a second extension, said memory storing a first indicator that is specific for said first file and that indicates the first extension is displayed in the user interface in a first style using a first displayed name, said memory storing a second indicator that indicates the second extension is displayed in the user interface in a second style using a second displayed name, said processor ~~detects~~to detect a conflict in naming the first file and the second file.

87. (original) A processing system as in claim 86 wherein if the first displayed name is the same as the second displayed name, then a conflict is detected.
88. (original) A processing system as in claim 86 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a conflict is detected.
89. (currently amended) A method of managing file extensions in a digital processing system with a user interface and a plurality of files, each file having a name that comprises a filename and an extension, said method comprising:
associating a file with an indicator which is user selectable for a subset of files in the plurality of files which have the same extension in said digital processing system and which indicates how to display an extension of the file; and
displaying a displayed name of the file in the user interface in a style determined by said indicator.

90. (original) A method as in claim 89 wherein the style is such that the displayed name contains the extension of the file only when said indicator is for showing the extension of the file.
91. (currently amended) A machine readable medium for use with a digital processing system which has a user interface and a plurality of files, each file having a name and an extension, said machine readable medium containing executable computer program instructions which when executed by said digital processing system causes said system to perform a method comprising:
associating a file with an indicator which is user selectable for a subset of files in the plurality of files which have the same extension in said digital processing system and which indicates how to display an extension of the file; and
displaying a displayed name of the file in the user interface in a style determined by said indicator.
92. (original) A medium as in claim 91 wherein the style is such that the displayed name contains the extension of the file only when said indicator is for showing the extension of the file.
93. (currently amended) A method of managing file extensions in a digital processing system with a user interface, said method comprising:
associating a first file with an indicator which is user selectable for a subset of a plurality of files in the digital processing system, said indicator indicating that first extensions of said subset of files are displayed in a user interface in a first

style which is different from a second style used to display at least a second file in said plurality of files, wherein said second file is not in said subset and has a second extension which is the same as at least one of said first extensions; and

displaying in said first style a first displayed name of said first file in the user interface.

94. (currently amended) A method as in claim 93 wherein said first style and said second style are selected from a set of styles, said set of styles ~~comprising~~ comprising:
- (a) showing an extension of a file being displayed; and
 - (b) hiding an extension of a file being displayed.
95. (currently amended) A method as in claim 94 further comprising:
- storing an option, wherein said option in one state indicates that unknown extensions are not extensions of files and said option on another state indicates that unknown extensions are extensions of ~~files~~ files; and
- determining an extension of a file using said option.
96. (currently amended) A machine readable medium for use with a digital processing system which has a user interface and a plurality of files, said machine readable medium containing executable computer program instructions which when executed by said digital processing system causes said system to perform a method comprising:
- associating a first file with an indicator which is user selectable for a subset of a plurality of files in the digital processing system, said indicator indicating that first extensions of said subset of files are displayed in a user interface in a first

style which is different from a second style used to display at least a second file in said plurality of files, wherein said second file is not in said subset and has a second extension which is the same as at least one of said first extensions; and
displaying in said first style a first displayed name of said first file in the user interface.

97. (currently amended) A ~~media~~-medium as in claim 96 wherein said first style and said second style are selected from a set of styles, said set of styles ~~comprising~~ comprising:
(a) showing an extension of a file being displayed; and
(b) hiding an extension of a file being displayed.
98. (currently amended) A ~~media~~-medium as in claim 97 wherein the method further comprises:
storing an option, wherein said option in one state indicates that unknown extensions are not extensions of files and said option on another state indicates that unknown extensions are extensions of ~~files~~; files; and
determining an extension of a file using said option.
99. (new) A method as in claim 1 wherein the indicator is not a part of a name of the file.
100. (new) A media as in claim 22 wherein the indicator is not a part of a name of the file.
101. (new) A system as in claim 43 wherein the indicator is not a part of a name of the file.

- 102. (new) A system as in claim 64 wherein the indicator is not a part of a name of the file.
- 103. (new) A method as in claim 89 wherein the indicator is not a part of a name of the file.
- 104. (new) A medium as in claim 91 wherein the indicator is not a part of a name of the file.
- 105. (new) A method as in claim 93 wherein the indicator is not a part of a name of the file.
- 106. (new) A medium as in claim 96 wherein the indicator is not a part of a name of the file.
- 107. (new) A system as in claim 86 wherein the first indicator is not a part of a name of the first file.